



**Escuela Superior
de Ingeniería y Tecnología**
Universidad de La Laguna

Trabajo de Fin de Grado

Aplicación de Edición de Imágenes con Inteligencia Artificial para la Automatización de Fotos Personalizadas

*Artificial Intelligence Image Editing Application for
Personalized Photo Automation*

Facundo José García Gallo

La Laguna, 23 de mayo de 2024

Dña. **María Elena Sánchez Nielsen**, profesora Titular de Universidad adscrita al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutora

C E R T I F I C A

Que la presente memoria titulada:

"Aplicación de Edición de Imágenes con Inteligencia Artificial para la Automatización de Fotos Personalizadas"

ha sido realizada bajo su dirección por D. **Facundo José García Gallo**.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 23 de mayo de 2024

Agradecimientos

Al finalizar lo que es probablemente la etapa más importante de mi vida, quiero expresar mi agradecimiento por todo lo que me han brindado en estos últimos cuatro años.

A mis profesores, gracias por compartir sus conocimientos y por motivarme a ser mejor cada día.

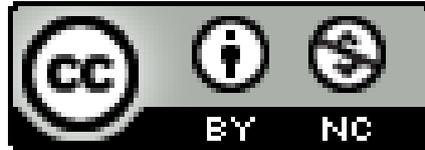
A mi tutora, por ser la guía de este último proyecto. Gracias por su paciencia y apoyo para completar esta etapa final.

A todos mis amigos, compañeros inseparables en este viaje. Gracias por los momentos compartidos y por hacer que estos años fueran inolvidables. Juntos hemos superado desafíos y hemos celebrado éxitos.

Finalmente, agradezco enormemente a toda mi familia. Mis padres son mis raíces, la base de todos mis valores y sentimientos; sin ellos, no sería la persona que soy hoy. Mi hermana, aunque sea la menor de los dos, es mi ejemplo a seguir en todo, mi referente. Mis abuelos, aún estando lejos, han sido una motivación inmensa en toda esta odisea y siempre han encontrado las palabras precisas para tranquilizarme. Max, esa bolita de pelo y felicidad que me alegra todas las mañanas. Y en general, a todas las personas que se han cruzado a lo largo de mi vida, haciendo que crezca cada día un poquito más.

Gracias.

Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-
NoComercial 4.0 Internacional.

Resumen

El presente Trabajo de Fin de Grado tiene como objetivo principal detallar el proceso llevado a cabo para la implementación de una aplicación destinada a la edición de imágenes mediante técnicas de inteligencia artificial, con un enfoque específico en la detección de rostros. Esta aplicación ofrece diversas opciones de edición, que van desde la aplicación de formatos y características propias de distintas plataformas o procesos burocráticos, hasta la aplicación de filtros y ajustes básicos que permiten modificar la imagen con propósitos artísticos o administrativos. Más concretamente, se implementaron tres tipos de formatos: el formato que establece las medidas necesarias para cumplimentar los requisitos del Documento Nacional de Identidad, el formato para aportar imágenes personales en áreas laborales como LinkedIn o currículums y el formato que establece medidas para redes sociales como Instagram. Por otro lado, se incluyen varios filtros implementados, tales como blanco y negro, frío, cálido, pintura al óleo, modo retrato y extracción del fondo.

Palabras clave: Inteligencia artificial, Machine Learning, Detección de rostros, Edición de Imágenes, Formatos de Imágenes, Filtros para Imágenes

Abstract

The main objective of this Final Project is to detail the process carried out for the implementation of an application aimed at image editing using artificial intelligence techniques, with a specific focus on face detection. This application offers various editing options, ranging from applying formats and characteristics specific to different platforms or bureaucratic processes to applying filters and basic adjustments that allow for image modification for artistic or administrative purposes. More specifically, three types of formats were implemented: the format that establishes the necessary measurements to meet the requirements for the National Identity Document, the format for providing personal images in professional areas such as LinkedIn or resumes and the format that sets measurements for social media such as Instagram. Additionally, several filters were implemented, including black and white, cold, warm, oil painting, portrait mode, and background removal.

Keywords: Artificial Intelligence, Machine Learning, Face Detection, Image Editing, Image Formats, Image Filters

Índice general

1. Introducción	1
1.1. Objetivos	1
1.1.1. Objetivos no funcionales	1
1.1.2. Objetivos funcionales	1
1.2. Ámbito y alcance	3
1.3. Metodología	4
2. Estado del arte	5
2.1. Implementaciones similares	6
3. Diseño	8
3.1. Propuesta de diseño lógico	8
3.2. Wireframe de la interfaz	9
4. Desarrollo	11
4.1. Esquema estructural lógico	11
4.2. Tecnologías usadas	11
4.3. Diseño de la página	13
4.4. Base de Datos	14
4.5. Backend	17
4.5.1. Conexión con la base de datos	17
4.5.2. OpenCV	18
4.5.3. Formatos	18
4.5.4. Filtros	19
4.5.5. Autenticación por token	20
4.6. Frontend	21
4.6.1. Páginas	21
4.6.2. Componentización	25
4.6.3. Almacenamiento local	25
4.7. Despliegue	26
4.7.1. Despliegue de la Base de Datos	26
4.7.2. Despliegue del Backend	27
4.7.3. Despliegue del Frontend	27
5. Evaluación Experimental	28
5.1. Diferencias y similitudes entre lo previsto y lo logrado	28
5.2. Ampliación de lo implementado	29
5.3. Análisis de acierto y error de detección de rostros	31

5.4. Pruebas de usuario	35
5.4.1. Información referente al usuario 1	38
5.4.2. Información referente al usuario 2	39
5.4.3. Información referente al usuario 3	40
5.4.4. Información referente al usuario 4	41
5.4.5. Información referente al usuario 5	42
5.4.6. Interpretación de las pruebas de usuario	43
6. Conclusiones y problemas encontrados	45
6.1. Límites	45
6.2. Problemas encontrados	45
6.3. Relación con materias cursadas	46
7. Summary and Conclusions	47
7.1. Limits	47
7.2. Problems found	47
7.3. Relationship with subjects studied	48
8. Presupuesto	49

Índice de Figuras

2.1. Ejemplo de manipulación de imágenes con Photoshop extraída de vídeo de ejemplo [8]	7
2.2. Ejemplo de manipulación de imágenes con FaceApp	7
3.1. Representación visual de la estructura lógica del proyecto	8
3.2. Wireframe de la interfaz	9
4.1. Colores usados	13
4.2. Logos	13
4.3. Modelo de datos de la base de datos	14
4.4. Definición de la tabla de usuarios	15
4.5. Definición de la tabla de imágenes	16
4.6. Definición del disparador que controla la inserción de imágenes	17
4.7. Conexión a la base de datos desde la API	17
4.8. Página principal	22
4.9. Página para la selección de imagen	22
4.10. Página para la selección del formato	23
4.11. Página para la selección del filtro	23
4.12. Página para la previsualización de la imagen	24
4.13. Página para la exportación de la imagen	24
5.1. Conexión a la base de datos desde la API	30
5.2. Conexión a la base de datos desde la API	30
5.3. Verdadero positivo (izquierda) frente a falso positivo (derecha)	31
5.4. Verdadero negativo (izquierda) frente a falso negativo (derecha)	32
5.5. Análisis del índice de acierto de detección de rostros para los verdaderos positivos	32
5.6. Análisis del índice de acierto de detección de rostros para los verdaderos negativos	33
5.7. Análisis de la métrica utilizada	34
5.8. Gráfica que representa la media por pregunta por criterio	44

Índice de Tablas

5.1. Pruebas de usuario	37
5.2. Respuestas del Usuario 1	38
5.3. Respuestas del Usuario 1	39
5.4. Respuestas del Usuario 2	39
5.5. Respuestas del Usuario 3	40
5.6. Respuestas del Usuario 4	41
5.7. Respuestas del Usuario 5	42
8.1. Resumen del presupuesto	49

Capítulo 1

Introducción

1.1. Objetivos

A lo largo de la siguiente sección se detallarán aquellos aspectos que se implementaron, abordando desde cuestiones más teóricas como análisis o evaluaciones hasta aspectos prácticos como los formatos o filtros que se implementaron. En capítulos posteriores se profundizará más en la manera de implementar estos objetivos y las dificultades encontradas. Los objetivos específicos que se abordaron durante la implementación del proyecto se pueden dividir en dos grandes grupos: los objetivos funcionales y los objetivos no funcionales.

1.1.1. Objetivos no funcionales

Comenzando con los objetivos no funcionales tenemos, en primer lugar, la realización de un análisis de precisión de la detección de rostros, con esto se logra evaluar el porcentaje de acierto que tiene el programa respecto a diferentes entradas de imágenes. Este estudio, al realizarse posterior a la implementación final, ayuda a deducir si se deben realizar cambios en la aplicación para lograr mejores resultados. Por otro lado, se debe realizar pruebas de usuario, un estudio junto con usuarios reales para detectar cualquier aspecto que necesite modificación o mejora. Estas pruebas permitirán evaluar la eficacia y la satisfacción de los usuarios con la aplicación.

1.1.2. Objetivos funcionales

Ya pasando con los objetivos funcionales tenemos: la realización de una aplicación dividida en tres capas [1], esto aporta diferentes ventajas como la modularización, la escalabilidad o la facilidad de desarrollo. Estos tres niveles se dividen en:

- **Base de Datos:** desarrollo de una estructura que permita el almacenamiento de distintas entidades relacionadas con la plataforma. Más concretamente se deben representar los usuarios y las imágenes asociadas a esos usuarios, estableciendo un límite de cinco imágenes por usuario para no exceder la capacidad de la base de datos.
- **Backend:** definición de una capa que permita la interacción con la base de datos para poder realizar operaciones de inserción, eliminación, obtención y modificación sobre las diferentes entidades de la base de datos. Este nivel también debe soportar todas las operaciones de edición de imágenes que veremos a continuación.
- **Frontend:** realización de una interfaz de usuario intuitiva y atractiva, que permita a los usuarios interactuar fácilmente con la aplicación para realizar las ediciones deseadas sobre las imágenes.

Siguiendo con los objetivos prácticos, en segundo lugar tenemos la realización de diferentes formatos que modifican imágenes adaptándolas a requisitos que se pueden encontrar en el día a día.

- **Formato DNI:** este formato modifica la imagen para que cumpla con las medidas establecidas para la tramitación de documentos oficiales, como por ejemplo la renovación del DNI. Esta implementación debe recibir una imagen de una única persona mirando de manera frontal a la cámara y con suficiente margen para poder realizar los recortes necesarios. También se tiene en cuenta la distancia a la que se encuentra la persona de la cámara. Como resultado se obtendrá la imagen recortada a color y con un fondo de color blanco sólido.
- **Formato Instagram o red social:** este formato realiza los cambios necesarios para que la imagen cumpla con los requisitos de distintas redes sociales. Para ello se introduce una imagen y la aplicación detecta la aparición de personas en la misma, aplicando los márgenes suficientes para que todas las personas aparezcan en el resultado final. La proporción de la imagen resultante es 1:1, por lo que tiene el mismo alto que ancho.
- **Formato currículum:** este formato detecta la inclusión de una única persona en la imagen, que debe mirar de manera frontal a la cámara. Como resultado se obtiene una imagen con proporción 1:1 y como objeto principal, la persona.

Para finalizar, tenemos la implementación de diferentes filtros que generan efectos visuales sobre las imágenes.

- Blanco y Negro: transforma la imagen original a escala de grises, eliminando colores.
- Filtro frío: enfatiza colores como el azul o verde, generando un ambiente amplio y tranquilo, con sensación de serenidad y distanciamiento.
- Filtro cálido: enfatiza colores como el rojo o amarillo, generando un ambiente acogedor y cercano, con sensación de confort y cercanía.
- Pintura al óleo: aplica efectos que simulan la textura de un cuadro pintado al óleo para obtener un aspecto artístico.
- Modo retrato: separa el objeto principal del fondo y aplica un efecto de desenfoque al fondo de la imagen, ayudando a resaltar el objeto principal y así crear una sensación de profundidad.
- Eliminación de fondo: separa el objeto principal del fondo y sustituye este último por un color blanco sólido.

1.2. **Ámbito y alcance**

La aplicación se puede emplear en cualquier ámbito que lo precise, ante cualquier persona que requiera de sus funcionalidades, desde adolescentes hasta personas mayores. Sin embargo se pueden detallar casos donde se puede tener más en cuenta la aplicación.

- **Integración en plataformas de redes sociales**, dando la comodidad al usuario de poder editar su imagen antes de compartirla en las redes.
- **Complementación en ambientes profesionales**, donde se requiera de formatos específicos para documentos oficiales, la aplicación podría utilizarse para el procesamiento y estandarización de imágenes.
- **Unificación con redes sociales orientadas al uso empresarial**, logrando que los usuarios tengan un área común donde crear su perfil corporativo y editar las imágenes asociadas al mismo.

Continuando con el alcance, podemos decir que la aplicación realizada tiene como objetivo dar una solución versátil y accesible para la edición de imágenes, sin enfocarse en un público concreto, tal y como se mencionó anteriormente. A continuación se detallan aquellos conjuntos de públicos que podrían estar más interesados en el uso de la plataforma.

- **Personas con necesidad de edición de imágenes**, al estar diseñado de manera intuitiva, se hace adecuado su uso para personas de cualquier edad que quieran editar imágenes fácilmente sin necesidad de poseer conocimientos técnicos en edición de imágenes.
- **Personas que se dedican a las redes sociales**, la aplicación ofrece la posibilidad de optimizar y adaptar las imágenes para su publicación en diferentes plataformas, garantizando una apariencia profesional y atractiva.
- **Trabajadores administrativos**, donde el uso de esta herramienta puede ser útil para mejorar la uniformidad de las imágenes utilizadas, aportando los formatos oficiales que se demandan en cada caso.
- **Personas que desean crear currículums**, la aplicación permite crear el formato ideal para imágenes que se desean colocar en currículums.
- **Personas que desean crear un perfil profesional de trabajo**, del mismo modo que con la elaboración de currículums, aquellas personas que aspiran a tener una presencia profesional en línea pueden beneficiarse de la aplicación.

1.3. Metodología

Durante la implementación del Trabajo Fin de Grado, se optó por una metodología más flexible, inspirada en la metodología ágil Scrum [2], pero adaptada a las necesidades del contexto. En lugar de seguir un camino estrictamente definido, se organizó el trabajo en etapas que permitieron mostrar avances periódicos a la tutora. En cada reunión, se mostró a la tutora el estado del proyecto y se aprovechó al máximo el tiempo para realizar preguntas sobre la implementación y así hacer ajustes y mejoras. Esta interacción, aunque no fue tan frecuente como en un entorno ágil típico, permitió una colaboración efectiva. La adaptación de esta metodología al contexto particular fue esencial, ya que la tutora desempeñaba el papel de *cliente* en el contexto académico. Su retroalimentación actuaba como guía para el desarrollo del proyecto, asegurando que cumpliera con las expectativas.

Capítulo 2

Estado del arte

La detección de rostros es la capacidad que tiene un sistema informático para identificar y localizar el rostro de una persona en una imagen o vídeo. En la actualidad, la detección de rostros se ha consolidado como una herramienta crucial en el ámbito de la edición de imágenes, facilitando y mejorando considerablemente la precisión y calidad de las modificaciones en retratos. Entre las técnicas para la detección de rostros que más se usan hoy en día se encuentran las siguientes:

- **Los clasificadores en cascada de Haar** [3], es una técnica tradicional que utiliza características Haar para identificar posibles rostros en una imagen. Las características de Haar son patrones simples de apariencia visual en una imagen en escala de grises que se utilizan como elementos para detectar objetos en la imagen. Esta técnica es rápida y sencilla de implementar, pero su precisión es limitada y es sensible a variaciones en iluminación y pose.
- **Los Histogramas de Gradientes Orientados** [4], se basan en la distribución de gradientes en una imagen, es decir, cada píxel tendrá asociado el cálculo de la magnitud y la dirección del gradiente. Esta técnica es robusta frente a cambios en iluminación y pose, puede ser utilizada en aplicaciones en tiempo real. Sin embargo, aunque es una técnica bastante eficiente, no es tan precisa como las redes neuronales convolucionales. Normalmente son complementados con un clasificador para mejorar su rendimiento.
- **Las redes neuronales convolucionales (CNN)** [5], representan un avance significativo en la detección de rostros. Estas redes aplican filtros convolucionales para extraer características jerárquicas de las imágenes, es decir, en las primeras capas se detectan propiedades sencillas, mientras que, a medida que se profundiza a través de las

capas, se detectan formas más complejas, hasta llegar al rostro de la persona, ofreciendo alta precisión y la capacidad de manejar variaciones en iluminación, pose y oclusión. No obstante, las CNN son computacionalmente intensivas y requieren grandes cantidades de datos para el entrenamiento.

- **Las Redes Neuronales de Regiones Propuestas** [6], generan regiones de interés en una imagen, donde es probable que se encuentren los rostros. Estas regiones propuestas luego se pasan a través de una red neuronal más compleja para la clasificación y la refinación de los límites de los rostros detectados. Tienen la ventaja de obtener resultados excelentes pero la contrapartida de tener una complejidad elevada. Actualmente se ha mejorado esta técnica para lograr mejoras en la velocidad y la precisión, llegando a soluciones como Fast R-CNN o Faster R-CNN.

En este proyecto se realiza la detección de rostros mediante la utilización de clasificadores en cascada de Haar. Para ello se empleó *OpenCV* [7], una librería de código abierto especializada en la visión por computador. Primero se realiza un escaneo mediante una región definida, denominada como *ventana*. Esta ventana recorre la imagen aplicando el clasificador a cada posición. Estos clasificadores son los que determinan si se trata del objeto que se quiere detectar o no, que en este caso se trata de un rostro. Además, el tamaño de la ventana es variable para determinar si existen diferentes rostros de distintos tamaños.

2.1. Implementaciones similares

Adobe Photoshop es ampliamente conocido como la aplicación perfecta para la edición de imágenes, la integración de técnicas de detección facial ha revolucionado la forma en que los usuarios pueden manipular sus imágenes. La función *Face-Aware Liquify* permite a los usuarios realizar ajustes detallados en características faciales específicas, como los ojos, la nariz, la boca y la estructura del rostro, con una precisión casi perfecta. Esta herramienta identifica automáticamente las características faciales y proporciona controles que permiten realizar modificaciones naturales y realistas. A continuación se puede ver un ejemplo de edición de rostros mediante esta técnica de *Photoshop*, donde es inapreciable la diferencia de la separación de los ojos si observamos cada imagen por separado.



Figura 2.1: Ejemplo de manipulación de imágenes con Photoshop extraída de vídeo de ejemplo [8]

FaceApp, por otro lado, es una aplicación para teléfonos móviles que ha tenido cada vez más repercusión gracias a sus impresionantes capacidades de transformación facial. Utilizando algoritmos avanzados de detección de rostros y aprendizaje automático, *FaceApp* permite a los usuarios aplicar una variedad de efectos y filtros a los rostros de manera automática y sorprendentemente realista. Entre sus funciones más populares se encuentran el cambio de edad, que permite a los usuarios ver versiones más jóvenes o más viejas de sí mismos, y la transformación de género, que aplica características masculinas o femeninas al rostro del usuario. Además, *FaceApp* ofrece opciones para cambiar peinados, agregar sonrisas y aplicar maquillaje virtual, todo con un nivel de detalle y realismo increíble.



Figura 2.2: Ejemplo de manipulación de imágenes con FaceApp

Capítulo 3

Diseño

En este pequeño análisis se pretende evaluar algunas propuestas de diseño que posteriormente se compararán con el trabajo realizado. Todas estas propuestas se realizaron previamente al comienzo del desarrollo.

3.1. Propuesta de diseño lógico

Después de consultar diferentes fuentes de información, se ha visto conveniente la utilización de *Python* [10] para el backend, para ello se usará en combinación con *Flask* [11], un framework muy intuitivo. Además, este framework destaca por su flexibilidad a la hora de integrar bibliotecas. Por otro lado, para la parte del frontend, se hará uso del framework *React* [12], que es una biblioteca de *JavaScript* [13] para construir interfaces de usuarios interactivas y dinámicas. Las características que llevaron a esta selección fueron la utilización de componentes reutilizables y *Virtual DOM*, que se verán en detalle más adelante. Finalmente se hará uso de un sistema gestor de base de datos relacional, concretamente se usará *PostgreSQL* [9], ya que es de código abierto y ofrece ventajas como el rendimiento óptimo y la escalabilidad.

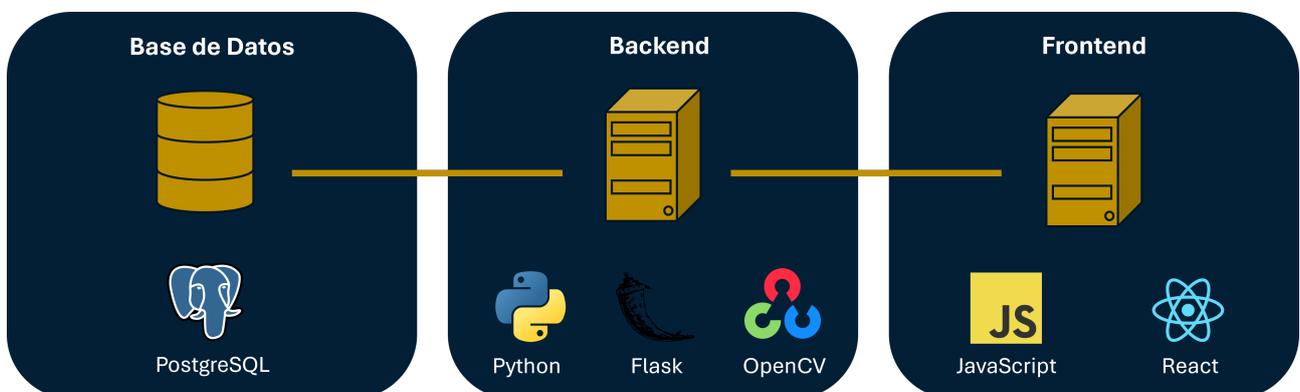


Figura 3.1: Representación visual de la estructura lógica del proyecto

3.2. Wireframe de la interfaz

En este apartado lo que se intenta es dar una visión general de la interfaz gráfica de la página gracias a la realización de un primer wireframe. Tal y como se puede ver a continuación, la página constará de varias secciones, cada una de ellas se pasarán a explicar brevemente a continuación. Sin embargo, cada página tendrá algo en común, una cabecera en la que se puede apreciar el logo de la plataforma, un menú deslizante en la zona izquierda y un indicador de workflow que especifica en qué sección del proceso se encuentra el usuario.

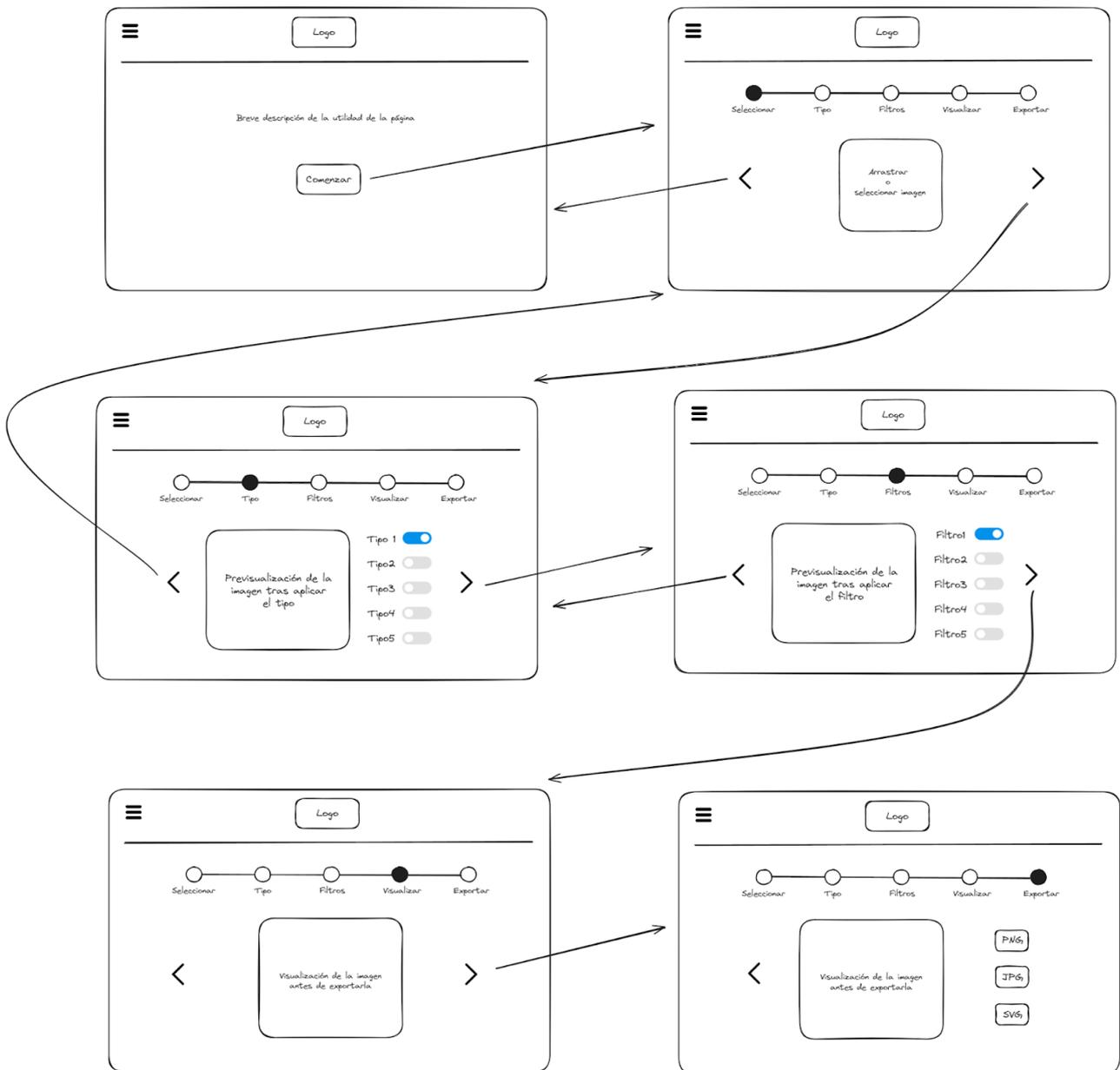


Figura 3.2: Wireframe de la interfaz

- La página principal será algo sencilla, en el cuerpo de la misma, aparece una descripción de la finalidad de la página, a modo de introducción. También se añaden ejemplos de resultados para atraer al usuario. Existirá un botón desde el que se podrá acceder a la página que se encarga de seleccionar la imagen, accediendo así al proceso de edición de las imágenes.
- La página de selección de imagen, desde donde el usuario puede seleccionar la imagen que desea editar. Al mismo tiempo, la página contará con dos botones para avanzar o retroceder en el proceso de edición.
- La pantalla de selección de formato, desde donde se pueden seleccionar diferentes versiones con características distintas según las diferentes plataformas donde se quieran publicar (Web empresarial/institucional, redes sociales, etc). Para ello se cuenta con un pequeño menú en el que se pueden seleccionar las distintas opciones.
- La página para la selección de filtros, donde se da la posibilidad de poder especificar los diferentes filtros que se pueden aplicar a la imagen. Para ello se cuenta con un pequeño menú en el que se pueden seleccionar las distintas opciones.
- La página de visualización del resultado, en esta página el usuario podrá observar de antemano el resultado que descargará en la próxima pantalla.
- La página del proceso de exportación, se podrán elegir las diferentes maneras de exportar el resultado final. Una vez exportada la imagen se podrá acceder nuevamente a la página principal de la aplicación.

Capítulo 4

Desarrollo

A continuación se hará un análisis del desarrollo implementado para lograr los objetivos acordados.

4.1. Esquema estructural lógico

En cuanto a la estructura lógica del proyecto, se siguieron las indicaciones que se analizaron en el estudio previo y se optó por dividir la aplicación en tres capas: base de datos, backend y frontend. Esta decisión se basa en varias razones que se consideraron importantes, en primer lugar, esta implementación en capas permite una mayor claridad y organización en el desarrollo del proyecto. Cada capa se encarga de funciones específicas, lo que facilita la gestión del código y la identificación de posibles problemas o mejoras. Además, al separar las responsabilidades en distintas capas, se promueve la modularidad, pilar básico en el sector del desarrollo de aplicaciones. Esto significa que se puede desarrollar y modificar cada capa de forma independiente, lo que agiliza el proceso de desarrollo y facilita la incorporación de nuevas funcionalidades o la corrección de errores. Otro punto a favor de esta división es la escalabilidad del proyecto, al tener una arquitectura bien definida, resulta más sencillo escalar el sistema para adaptarse a un mayor volumen de usuarios. Sin embargo, esto aporta una desventaja clara, la dificultad a la hora de conectar todas las partes y hacer que todo se comunique.

4.2. Tecnologías usadas

Para hablar de las tecnologías utilizadas en el proyecto, volvemos a dividir el esquema en las tres partes fundamentales: base de datos, backend y frontend.

Para la gestión de la base de datos, se ha empleado *PostgreSQL* como sistema gestor. *PostgreSQL* es un sistema de gestión de bases de datos relacional de código abierto y altamente potente. Ofrece una amplia gama de características avanzadas de las que se hablará en profundidad en la implementación de la propia base de datos. Su flexibilidad y escalabilidad

lo hacen ideal para aplicaciones de diversos tamaños y complejidades. Se decidió utilizar este gestor de base de datos porque significó una oportunidad para aprender en profundidad el comportamiento de este sistema ante el almacenamiento de imágenes, cuestión que no se había tratado hasta el momento.

En la capa del backend, se ha optado por utilizar *Python* en combinación con *Flask* como framework. *Python* es un lenguaje de programación de alto nivel e interpretado. *Flask*, por su parte, es un framework web minimalista y ligero para *Python*. Permite construir aplicaciones web de forma rápida y sencilla, proporcionando las herramientas necesarias para la creación de *APIs RESTful* y servicios web. La decisión de la utilización de esta combinación de tecnologías viene de querer profundizar más en cada una, ya que a lo largo de la carrera no se había tocado en profundidad ninguna de ellas.

Para el desarrollo del frontend, se han utilizado *JavaScript* y *React* como framework. *JavaScript* es un lenguaje de programación ampliamente utilizado en el desarrollo web. *React*, por otro lado, es una biblioteca de *JavaScript* de código abierto para construir interfaces de usuario. Se destaca por su enfoque en componentes reutilizables y su capacidad para gestionar el estado de la aplicación de manera eficiente, lo que lo hace ideal para el desarrollo de interfaces de usuario. Se optó por usar este framework por su principal ventaja, la utilización del *Virtual DOM*. Gracias a ello se logra una eficiencia en rendimiento notable al minimizar el costo de renderizado y actualizar únicamente los elementos del DOM que han cambiado. Esta eficiencia no solo mejora el rendimiento de la aplicación, sino que también reduce la carga en el navegador, lo que resulta en una experiencia de usuario más fluida y receptiva.

4.3. Diseño de la página

El diseño de la página se ha estructurado considerando una paleta de colores específica para favorecer la accesibilidad de todos los usuarios. Se han seleccionado cuatro colores principales: el azul oscuro, el amarillo anaranjado, el azul marino, y el blanco.

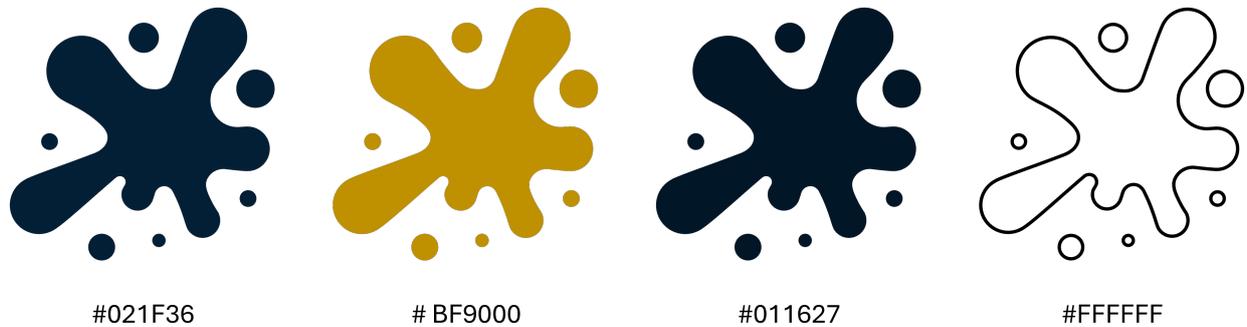


Figura 4.1: Colores usados

Se ha procurado obtener un contraste claro entre esta gama de colores para garantizar una experiencia visual óptima en todos los usuarios. Para ello, se han integrado estos colores en dos temas distintos: el tema oscuro y el tema claro. En relación al logo, se han empleado los mismos colores mencionados anteriormente.



Figura 4.2: Logos

El diseño de la página se caracteriza por un enfoque minimalista, donde se evitan detalles innecesarios. Esta elección busca generar una sensación de tranquilidad y sencillez en el usuario, permitiéndole centrarse en el contenido sin distracciones innecesarias.

4.4. Base de Datos

Tal y como se explicó anteriormente, el sistema gestor de bases de datos utilizado es *PostgreSQL*, una herramienta muy versátil que puede usarse ante todo tipo de proyectos, sin atender a la complejidad o a la escala. El modelo de datos usado es muy sencillo ya que la aplicación no requiere mucha complejidad hablando en términos de almacenamiento y gestión de datos. Para empezar solo se va a disponer de dos entidades, la del usuario (*users*) y la de la imagen (*images*).

Los atributos de la entidad *users* se describen a continuación:

- La clave primaria es el nombre de usuario (*username*), un campo que representa la identificación del usuario en la aplicación.
- El primer nombre del usuario (*first_name*).
- Los apellidos del usuario (*last_name*).
- El email asociado a ese usuario (*email*).
- La contraseña para poder acceder a la propia aplicación (*password*).

Los atributos de la entidad *images* aparecen a continuación:

- La clave primaria es el número secuencial que representa el identificador (*id*), es decir, un número secuencial que se asigna automáticamente y en orden ascendente. Esto aporta grandes beneficios como la unicidad o la integridad referencial.
- El usuario que posee la imagen (*username*).
- La propia imagen codificada (*image*).

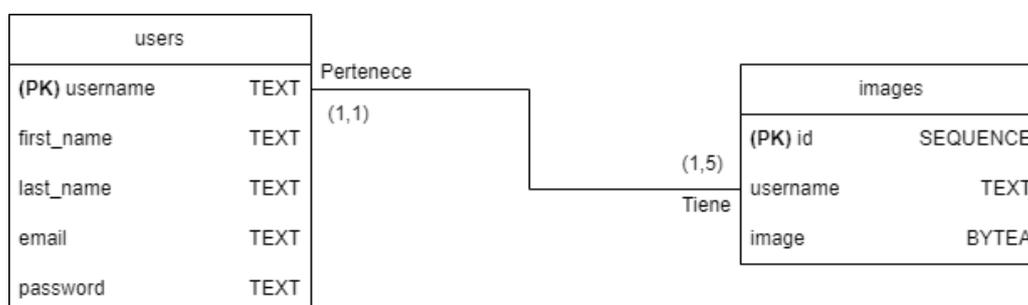


Figura 4.3: Modelo de datos de la base de datos

Antes de pasar a definir la creación de las tablas se debe explicar por qué se puso un límite en la relación entre ambas tablas. El almacenamiento de imágenes no es algo sencillo de gestionar, ya que no son tipos de datos simples, es por esto que la base de datos puede llegar a sobrecargarse si se tienen demasiadas imágenes almacenadas, aunque en el transcurso normal de la página web probablemente ningún usuario llegue a almacenar tantas imágenes como para sobrecargar la base de datos.

```
1 CREATE TABLE users (  
2   username TEXT PRIMARY KEY,  
3   first_name TEXT NOT NULL,  
4   last_name TEXT NOT NULL,  
5   email TEXT NOT NULL,  
6   password TEXT NOT NULL,  
7  
8   CONSTRAINT no_blank_spaces_username CHECK (username ~ '^[A-Za-z0-9]+$'),  
9   CONSTRAINT valid_email CHECK (email ~ '^[A-Za-z0-9._%]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$'),  
10  CONSTRAINT username_length CHECK (LENGTH(username) <= 20 AND LENGTH(username) > 0),  
11  CONSTRAINT first_name_length CHECK (LENGTH(first_name) <= 50),  
12  CONSTRAINT last_name_length CHECK (LENGTH(last_name) <= 50)  
13 );
```

Figura 4.4: Definición de la tabla de usuarios

Ya explicado el modelo de datos, podemos pasar a su implementación en *PostgreSQL*. Para empezar se creó la tabla *users*, con todos los atributos mencionados anteriormente, pero agregando distintas restricciones. La primera de las restricciones, *no_blank_spaces_username*, como su propio nombre indica, se trata del formato que se le debe dar al nombre de usuario para que sea válido, solo letras y números, sin ningún espacio. La siguiente restricción es *valid_email*, que se encarga de verificar que el email introducido en la tabla tiene el formato correcto: se comienza como mínimo con una letra, seguido del carácter “@”, a continuación deben aparecer uno o más caracteres, un punto y dos o más letras pertenecientes al dominio del correo.

Las últimas tres restricciones realizan funciones similares, la primera de ellas logra controlar las dimensiones del nombre de usuario, que los límites son: un carácter mínimo y veinte caracteres máximo. Mientras que las dos últimas restricciones controlan las dimensiones del nombre y el apellido del usuario, que ambos establecen un límite máximo de cincuenta caracteres.

```

1 CREATE TABLE images (
2   id TEXT NOT NULL PRIMARY KEY ,
3   username TEXT,
4   image BYTEA NOT NULL
5 );
6
7 ALTER TABLE images OWNER TO custom_fix;
8
9 ALTER TABLE ONLY images
10  ADD CONSTRAINT username_fkey
11  FOREIGN KEY (username)
12  REFERENCES users(username)
13  ON UPDATE CASCADE
14  ON DELETE CASCADE;
15
16 CREATE SEQUENCE images_id_seq
17  AS integer
18  START WITH 1
19  INCREMENT BY 1
20  NO MINVALUE
21  NO MAXVALUE
22  CACHE 1;
23
24 ALTER SEQUENCE images_id_seq OWNER TO custom_fix;
25
26 ALTER TABLE ONLY images ALTER COLUMN id SET DEFAULT nextval('images_id_seq'::regclass);

```

Figura 4.5: Definición de la tabla de imágenes

Pasando con la implementación de *images* podemos destacar la creación del número secuencial que distingue cada registro de la tabla, este atributo comienza por el número uno, incrementando en una unidad por cada tupla y no tiene límite máximo. También aparece la relación que existe entre esta tabla y la tabla *users*, gracias a la utilización de una clave foránea que referencia al atributo *username* de la tabla *users*.

Tal y como se explicó previamente, la inserción de imágenes en la base de datos debe tener un límite, y es por esto que se decidió crear un trigger o disparador para gestionar el número de imágenes que tiene cada usuario en la base de datos. Para ello se crea una función que hace un recuento del número de entradas que tiene asociadas cada usuario en la tabla *images*, si este número es mayor que cinco, se lanza una excepción, indicando que el usuario no puede tener más del límite de imágenes. A continuación aparece la definición del trigger, que se ejecuta ante cada inserción en la tabla, ejecutando la función definida anteriormente.

```

1 CREATE OR REPLACE FUNCTION limit_image_count()
2 RETURNS TRIGGER AS $$
3 BEGIN
4 IF (SELECT COUNT(*) FROM images WHERE username = NEW.username) >= 5 THEN
5 RAISE EXCEPTION 'No more than 5 images per user are allowed';
6 END IF;
7 RETURN NEW;
8 END;
9 $$ LANGUAGE plpgsql;
10
11 CREATE TRIGGER check_image_count
12 BEFORE INSERT ON images
13 FOR EACH ROW
14 EXECUTE FUNCTION limit_image_count();

```

Figura 4.6: Definición del disparador que controla la inserción de imágenes

4.5. Backend

El backend es la parte encargada de procesar la información de entrada y de salida, interactuar con la base de datos para obtener y almacenar los datos y el punto medio entre la parte visual de la aplicación, es decir, el frontend, con la base de datos.

4.5.1. Conexión con la base de datos

Para la conexión entre el backend de la aplicación y la base de datos descrita en *PostgreSQL*, se hizo uso de *Psycopg*, un adaptador del sistema gestor de la base de datos para *Python*, lenguaje que, como se explicó anteriormente, se usa en el backend de la aplicación. Este adaptador permite, entre otras, conectarse a una base de datos y ejecutar consultas *SQL*, acciones principales que deben realizarse en la aplicación.

```

1 import psycopg2
2
3 def get_db_connection():
4     try:
5         conn = psycopg2.connect(host='XXX',
6                                 database="XXX",
7                                 user="XXX",
8                                 password="XXX")
9     except psycopg2.Error as error:
10         print("Database connection failure", error)
11         raise error
12     except Exception as error:
13         print("An error has occurred", error)
14         raise error
15     return conn

```

Figura 4.7: Conexión a la base de datos desde la API

4.5.2. OpenCV

OpenCV es una librería de código abierto diseñada para ofrecer una amplia gama de funciones relacionadas con el procesamiento de imágenes y la visión por computador. Esta librería proporciona herramientas y algoritmos para realizar diversas tareas en el campo de la visión por computador, como el procesamiento de imágenes, la detección y seguimiento de objetos (entre ellos, los rostros) o la manipulación de imágenes añadiendo muchas funcionalidades.

Dentro del contexto de la aplicación, *OpenCV* se despliega para dos propósitos fundamentales: la detección de rostros y la edición de imágenes. En el caso de la detección de rostros, la implementación se basa en el uso de un modelo previamente entrenado, que opera como un clasificador en cascada. Esta técnica emplea archivos proporcionados por *OpenCV*, los cuales contienen ejemplos tanto positivos como negativos de rostros humanos. A través de la función *detectMultiScale*, estos clasificadores son aplicados para detectar rostros en las imágenes. Es importante destacar que antes de hacer la llamada a la función, la imagen debe ser convertida a escala de grises para tener mejores resultados, ya que con esto, se reduce la complejidad y el costo computacional de procesamiento, enfocándose en las características importantes de la imagen y eliminando variaciones de color irrelevantes, es decir, cambios en la iluminación y variaciones en los colores de la piel.

Por otro lado, para la edición de imágenes, *OpenCV* proporciona una amplia gama de funciones para manipular y procesar imágenes de diversas maneras. Esto incluye operaciones básicas como el cambio de tamaño, el recorte y la rotación de imágenes, así como técnicas más avanzadas como la corrección de color, el filtrado de imágenes, la superposición de imágenes y la eliminación de ruido.

4.5.3. Formatos

En el ámbito de la aplicación, un formato se refiere a un conjunto específico de características y requisitos que debe cumplir una imagen para adaptarse a ciertos estándares o necesidades particulares. Los formatos definidos incluyen el formato DNI, el formato Instagram y el formato currículum.

- El formato DNI primero determina el tamaño de la imagen, clasificándola en pequeña (<2.000px), mediana (>2.000px, <4.000px) y grande

(>4.000px). En base a ese tamaño, se detectan los rostros frontales a la cámara, aplicando los parámetros adecuados al tamaño de la imagen. En caso de solo detectar un rostro, se realizan los cálculos necesarios para obtener las dimensiones de la imagen resultante, recortando la imagen para que quede más alta que ancha. Finalmente, se extrae el fondo de la imagen, aplicando un color blanco sólido.

- El formato Instagram primero llama a la función que detecta personas, incluyendo tanto rostros (frontales y de perfil) como cuerpos enteros. En base a lo detectado, se calculan las dimensiones que maximizan el tamaño y que satisfacen la condición de que aparezcan todas las personas en la imagen. Hay que destacar que se recorta la imagen de manera que quede del mismo alto que ancho.
- El formato currículum primero determina el tamaño de la imagen, clasificándola en pequeña (<2.000px), mediana (>2.000px, <4.000px) y grande (>4.000px). En base a ese tamaño, se detectan los rostros frontales a la cámara. Si realmente se detectó un único rostro, se recorta la imagen de manera que quede del mismo alto que ancho y que maximice el tamaño de la imagen con el rostro en el centro.

4.5.4. Filtros

Los filtros disponibles en la aplicación cumplen distintas funciones y añaden efectos específicos a las imágenes para adaptarlas a diferentes necesidades y estilos visuales. A continuación se explicará la manera de lograr los resultados esperados.

- El filtro de blanco y negro transforma la imagen original a una escala de grises, eliminando los colores y resaltando los contrastes. Para lograr este efecto, se hizo uso del método *cvtColor* de la librería *OpenCV*.
- El filtro frío añade tonos azulados o verdosos a la imagen. Para conseguir este filtro se divide la imagen en tres canales de color: azul, verde y rojo, y luego se aumenta la intensidad del canal del color azul, añadiendo la sensación de frío, luego se vuelven a combinar los canales de color para obtener la imagen reconstruida con las modificaciones aplicadas.
- Por otro lado, el filtro cálido agrega tonos rojizos, anaranjados o amarillentos. Para este filtro se vuelve a realizar la misma operación

que para el filtro frío pero aumentando la intensidad del canal de color rojo.

- El filtro de pintura al óleo aplica efectos que simulan la textura y apariencia de una pintura creada con óleo sobre lienzo. Para este filtro se utiliza un método *oilPainting*, sin embargo, este no se encuentra en el paquete por defecto de *OpenCV*, sino dentro del módulo *cv2.xphoto* por lo que para poder utilizar esta función se requiere la instalación adicional de este módulo.
- El filtro de eliminar fondo utiliza técnicas avanzadas para eliminar el fondo de la imagen, dejando al sujeto principal en primer plano y el fondo reemplazado por un fondo blanco sólido. Para ello se hace uso del método *remove* de la librería *rembg*. Su implementación resultaba tan sencilla y daba tan buenos resultados que se priorizó sobre la opción de *OpenCV*, sacrificando tiempo de ejecución y mejorando los resultados.
- Finalmente, el filtro de retrato aplica un efecto de desenfoque al fondo de la imagen, mientras mantiene enfocado al sujeto principal en primer plano. Para este filtro nuevamente se hace uso del método *remove* de la librería *rembg* pero se aplican nuevas técnicas para lograr ese efecto de desenfoque. Concretamente lo que se realiza es lo siguiente: se realiza una copia de la imagen original, a una de ellas se le extrae el fondo de la imagen, mientras que a la otra se le aplica un efecto difuminado, finalmente se fusionan ambas imágenes mediante la técnica de máscara alfa, logrando una combinación suave e inapreciable. La técnica de máscara alfa consiste en determinar la opacidad de cada píxel en una imagen, donde los píxeles pertenecientes al fondo de la imagen tienen un valor cercano al cero y los píxeles correspondientes al sujeto principal tienen un valor máximo. Finalmente, se combina suavemente la imagen desenfocada con el sujeto principal, permitiendo que los píxeles del sujeto mantengan su opacidad completa mientras que los del fondo se mezclan gradualmente con el fondo difuminado.

4.5.5. Autenticación por token

La autenticación por token es un método de seguridad utilizado en aplicaciones web para verificar la identidad de un usuario. Consiste en la generación de un token único que se utiliza para identificar al usuario durante su sesión. Este token se incluye en peticiones realizadas por el

usuario y se verifica en el servidor para garantizar que el usuario tiene permiso para acceder a los recursos protegidos. Una de las principales ventajas de la autenticación por token es su capacidad para mejorar la seguridad de la aplicación.

Para ello, en el código, se definen dos funciones principales: `generate_token` y `verify_token`. La primera función, `generate_token`, se encarga de crear un token JWT (JSON Web Token) [14] para un usuario específico. Este token incluye información sobre el usuario y una fecha de expiración. La función `verify_token`, por otro lado, verifica la validez y la integridad de un token dado. Si el token es válido y no ha sido modificado, se extrae la información del usuario del token y se devuelve. En caso de que el token haya expirado o sea inválido, se genera un mensaje de error.

En el archivo `login.py`, que maneja las solicitudes de inicio de sesión de los usuarios, se utiliza la autenticación por token. Después de que un usuario envía sus credenciales de inicio de sesión (nombre de usuario y contraseña), el servidor verifica la validez de las credenciales y, si son correctas, genera un token para ese usuario, utilizando la función `generate_token`.

Luego, en aquellos métodos que precisen de la autenticación, se incluye un bloque de código para verificar la validez del token enviado por el cliente. Este bloque extrae el token de la cabecera de la solicitud, lo verifica utilizando la función `verify_token`, y luego compara el usuario asociado al token con el usuario que está intentando acceder al recurso protegido. Si hay una coincidencia, se permite el acceso al recurso. De lo contrario, se devuelve un mensaje de error indicando credenciales inválidas.

4.6. Frontend

Tal y como se describió anteriormente, el frontend de la aplicación está desarrollado en *JavaScript* y *React*. A continuación se detallarán aspectos más concretos de este nivel de abstracción.

4.6.1. Páginas

A continuación, se detallará el diseño final de las páginas que se analizaron previamente en el apartado de diseño, destacando sus características principales, su funcionalidad y su contribución al flujo general del usuario.

Comenzamos con la página principal en la que se puede apreciar el

acceso directo a la edición de imágenes. A su vez, se puede ver un pequeño texto en el que se detalla brevemente la finalidad de la aplicación y una serie de gifs que describen visualmente los resultados que se pueden obtener en la misma, tal y como se diseñó en el wireframe.



Figura 4.8: Página principal

A continuación aparecen las distintas pantallas desde las que se editan las imágenes, las pantallas del flujo natural de edición de imágenes. Estas pantallas tienen elementos comunes pero se dividen en cinco:

- La primera de ellas se trata de la pantalla de selección de la imagen, desde ella también se puede volver a la página principal o acceder a la siguiente pantalla de edición.

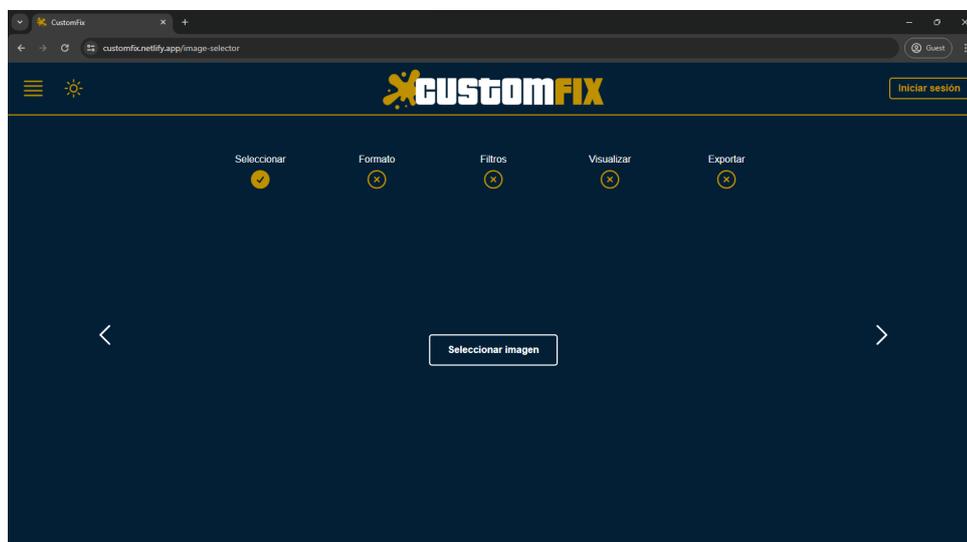


Figura 4.9: Página para la selección de imagen

- La segunda pantalla permite seleccionar el formato que se quiere aplicar sobre la imagen que se está editando, también se puede volver al paso anterior o pasar al siguiente paso, incluso sin seleccionar ningún formato. También aparece información sobre los distintos formatos, para que el usuario siempre tenga noción de lo que realiza cada opción.

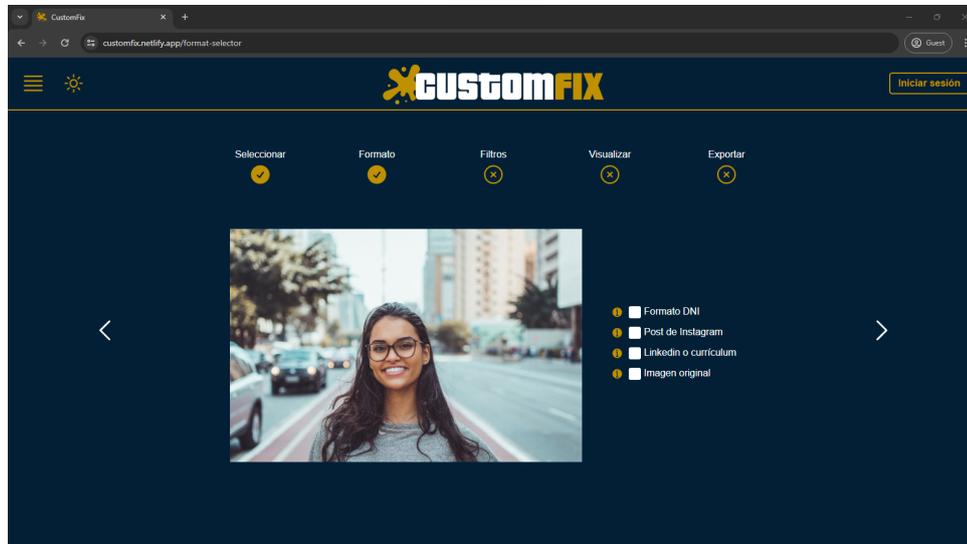


Figura 4.10: Página para la selección del formato

- La tercera pantalla del proceso de edición se trata de la pantalla que permite seleccionar un filtro para que se aplique sobre la imagen, del mismo modo que sucedía en la pantalla anterior, existen tarjetas con la información de cada filtro.

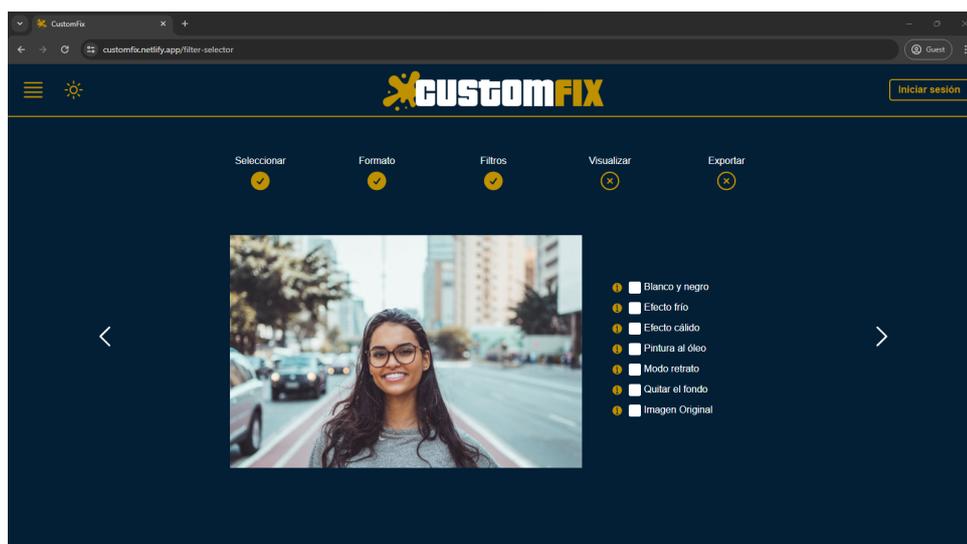


Figura 4.11: Página para la selección del filtro

- La cuarta pantalla es la pantalla de previsualización de la imagen, es una pantalla informativa, que tiene como finalidad mostrar al usuario el resultado que obtendrá en el siguiente paso.

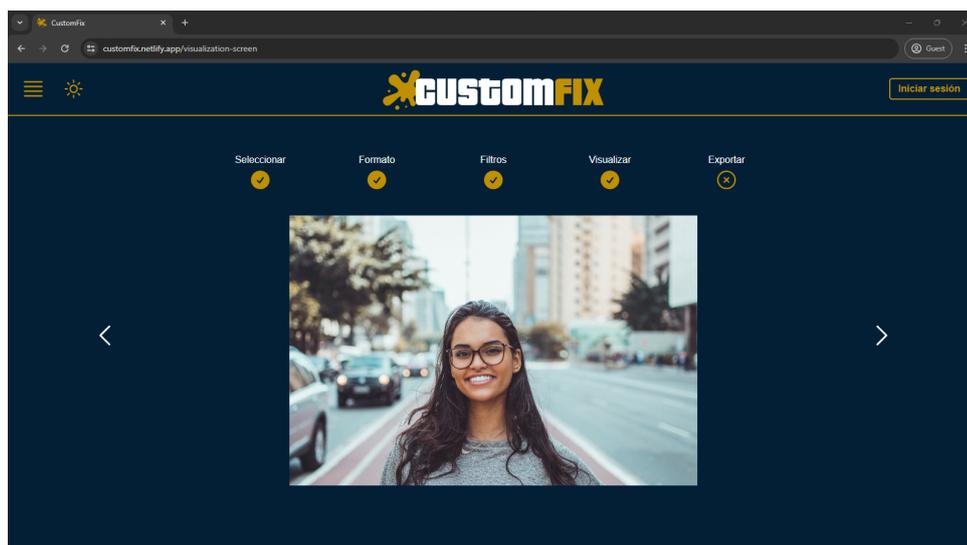


Figura 4.12: Página para la previsualización de la imagen

- En la última pantalla de las cinco, podemos encontrar las opciones de exportación de la imagen, donde aparecen distintos tipos de formatos.

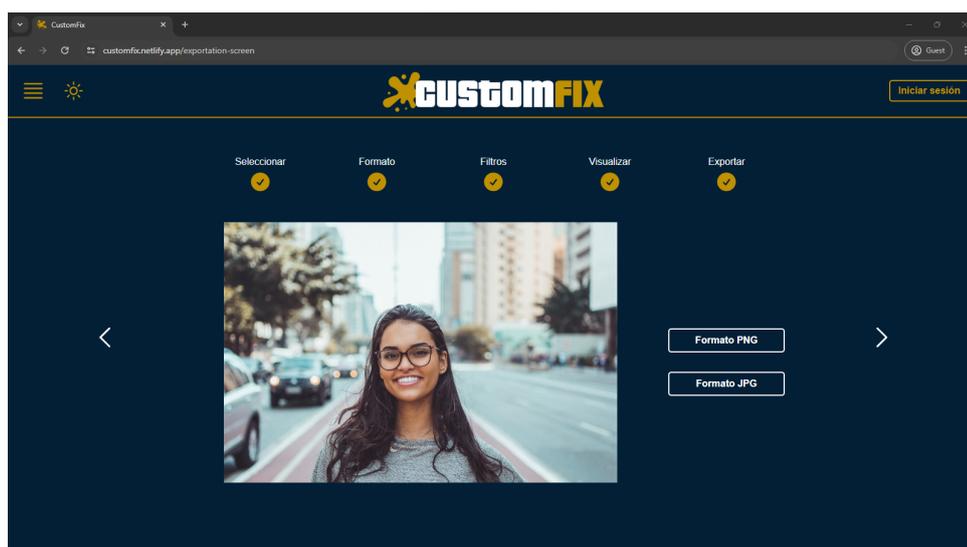


Figura 4.13: Página para la exportación de la imagen

Además, existen pantallas ajenas a todo el proceso de edición de la imagen, como por ejemplo la pantalla de inicio de sesión, donde el usuario puede autenticarse usando su nombre de usuario y su contraseña para poder obtener el beneficio de almacenar la imagen en la base de datos. Esta pantalla ofrece un acceso directo a la pantalla de registro por si se da el caso de que el usuario no tenga una cuenta creada.

La siguiente pantalla se trata de la pantalla de registro de usuario, introduciendo sus datos personales, puede crear una cuenta en la aplicación. Algunos campos tienen un formato específico, de manera que, si se introducen mal, se notifica al usuario. En caso de crear la cuenta satisfac-

toriamente, la página envía al usuario directamente a la pantalla de inicio de sesión para que pueda autenticarse.

La pantalla de contacto permite al desarrollador acercarse más al usuario de la página. Esta página aporta accesos directos a redes sociales como GitHub o LinkedIn y al correo personal para poder mantener contacto con aquellas personas que lo soliciten.

Por otro lado aparece la página en la que el desarrollador se da a conocer un poco, la página Sobre Mí, en la que se detalla un poco el paso por la universidad y las motivaciones que dieron lugar a los logros durante el periodo académico.

Una de las pantallas más importantes es la pantalla del perfil de usuario, en la que, al estar iniciada la sesión, el usuario puede consultar y modificar información personal e imágenes almacenadas en la base de datos de la aplicación. Al mismo tiempo se ofrece un acceso directo para la edición de las imágenes.

4.6.2. Componentización

La componentización en React es un enfoque fundamental que facilita la construcción de interfaces de usuario dinámicas y escalables. Estos componentes son bloques de código reutilizables que encapsulan el comportamiento de una parte específica de la interfaz de usuario. La principal ventaja de esta característica es la reutilización del código, ya que los componentes pueden ser fácilmente integrados y utilizados en diferentes partes de la aplicación. Además, al dividir la interfaz de usuario en componentes más pequeños, se facilita la depuración y la modificación de código, ya que cada componente es más fácil de entender y mantener por separado. En la aplicación existen muchos ejemplos de componentización, como es el caso de la cabecera, que se reutiliza en todas las páginas de la aplicación pero solo está definido una vez. Otro buen ejemplo es el de la barra que indica al usuario en qué paso de la edición de imagen se encuentra, que, combinándolo con los estados de React también se trata de un componente que se reutiliza en cada pantalla.

4.6.3. Almacenamiento local

El almacenamiento local o local storage es una característica esencial de los navegadores web que permite guardar datos de forma persistente en el navegador del usuario. Esto significa que los datos almacenados

permanecen disponibles incluso después de cerrar el navegador. Esto es particularmente útil en aplicaciones que requieren autenticación de usuario, ya que permite que los usuarios permanezcan conectados sin tener que iniciar sesión cada vez que entra a la página. Además, el almacenamiento local proporciona un acceso rápido a los datos almacenados, lo que mejora la experiencia del usuario al reducir los tiempos de carga y procesamiento. Además, se quiso dar otro enfoque más al almacenamiento local y, aunque las imágenes editadas no se almacenan de la misma manera que la información del usuario, se utilizó el almacenamiento local para almacenar temporalmente las imágenes que el usuario está editando. Esto asegura que las imágenes no se pierdan si el usuario abandona la página accidentalmente, sin embargo, estas imágenes se eliminan del almacenamiento local una vez que el usuario finaliza su sesión o recarga la página.

4.7. Despliegue

4.7.1. Despliegue de la Base de Datos

El despliegue de la base de datos consiste en desplazar todo lo implementado en local hacia la nube para poder ser accedido desde cualquier lugar. Para el despliegue lo que se realizó fue la implementación de diferentes scripts que automatizan la creación de la base de datos, por lo que su despliegue se hizo más sencillo.

Al comienzo se intentó realizar el despliegue en conjunto con el backend en la plataforma Heroku, pero las limitaciones de almacenamiento llevaron a descartar esta opción. Después de investigar por un tiempo y realizar muchas pruebas en distintas plataformas, se llegó a lo que en un futuro sería la solución final de despliegue, tanto para la base de datos como para el backend, Google Cloud Platform [15]. Esta plataforma otorga mucha libertad al desarrollador dado que la gran mayoría de los recursos son personalizables, aunque la modificación de estas características supone un aumento de los costos asociados.

Lo siguiente fue la creación de una instancia nombrada *customfix*. Se configuró con los parámetros necesarios para el acceso seguro mediante usuario y contraseña. Para ello se creó un usuario desde la pestaña de conexiones y finalmente se accedió a la base de datos y se crearon las tablas pertenecientes al modelo de datos descrito anteriormente.

4.7.2. Despliegue del Backend

Del mismo modo que la base de datos, el backend debía desplegarse para que pueda ser accesible desde cualquier lugar. Para ello, nuevamente se vio conveniente el uso de Google Cloud Platform. Esta vez se hizo uso de Google App Engine, que otorga al usuario una máquina virtual donde se tiene absoluta libertad para instalar cualquier tipo de librerías y aplicaciones. Previamente se tuvo que crear un fichero donde se especifiquen todos los requisitos para crear la aplicación. Para el despliegue final de la aplicación se hizo uso del comando `gcloud app deploy`, el cual sube los paquetes de la aplicación al servicio de Google Cloud Platform y configura de manera sencilla y automática la aplicación en el entorno de ejecución de Google App Engine.

4.7.3. Despliegue del Frontend

Del mismo modo que sucedió con la base de datos y el backend, el despliegue del frontend era inevitable. Para ello, esta vez se hizo uso de otra plataforma dedicada al despliegue de todo tipo de interfaces de páginas web, Netlify. La decisión del uso de esta plataforma ha sido muy acertada porque en muy poco tiempo permitió el despliegue del frontend de la aplicación, ofreciendo accesibilidad global para que la aplicación web esté disponible en todo momento y desde cualquier lugar, lo que amplía significativamente el alcance.

La integración con GitHub para el despliegue continuo es otra ventaja destacada. Cada vez que se realiza una actualización en el repositorio, Netlify se encarga automáticamente de desplegar la versión más reciente del frontend. Esta automatización no solo permite mantener la aplicación siempre actualizada, sino que también agiliza el proceso de desarrollo al eliminar la necesidad de realizar despliegues manuales.

Para configurar el despliegue en Netlify, se tuvo que definir algunas variables clave, como el repositorio de GitHub, la rama principal y la carpeta base del frontend. Estas configuraciones han permitido establecer un flujo de trabajo coherente y asegurar que el despliegue refleje siempre la versión más reciente del código.

Capítulo 5

Evaluación Experimental

5.1. Diferencias y similitudes entre lo previsto y lo logrado

En este apartado se van a detallar las diferencias entre el estudio realizado previamente al desarrollo de la aplicación y lo conseguido en la misma.

- En primer lugar, se puede destacar que la idea de la estructuración del proyecto se mantuvo tal como se había previsto en el análisis inicial, es decir, se logró establecer una clara separación entre la base de datos, el backend y el frontend, lo que facilitó la gestión y el desarrollo de cada capa de manera independiente.
- En cuanto a las tecnologías utilizadas, se pudo cumplir con la selección prevista: *PostgreSQL* para la gestión de la base de datos, *Flask* y *Python* para el desarrollo del backend, y *React* con *JavaScript* para la construcción del frontend. Esta elección se mostró acertada y permitió desarrollar la aplicación de manera eficiente y efectiva.
- En lo que respecta al diseño de la interfaz, se logró implementar todas las características previstas sin mayores cambios. Además, se pudieron agregar páginas nuevas, lo que contribuyó a mejorar la consistencia y estructura general de la aplicación, cumpliendo así con los objetivos establecidos en el análisis inicial.
- Se utilizó *OpenCV* para todas las implementaciones, excepto una, la extracción del fondo de las imágenes. Durante las primeras iteraciones del proyecto, se tenía la idea de utilizar *OpenCV* para todas las operaciones de procesamiento de imágenes. Sin embargo, se encontraron aspectos en los que no se cumplía con las expectativas, más concretamente en la extracción del fondo. Aunque su ejecución era

rápida, la calidad no estaba a la altura. Después de evaluar algunas opciones, se decidió probar con la librería *rembg* explicada anteriormente, diseñada específicamente para eliminar fondos de imágenes. Aunque tomaba un poco más de tiempo, los resultados eran mucho mejores. Se optó por esta alternativa, priorizando la calidad sobre la velocidad.

- Probablemente el punto que más difiere entre el análisis previo y lo realizado es el aspecto del despliegue, debido a que se detectaron algunos imprevistos. La implementación del frontend en *Netlify* fue bastante sencilla, tal como se había planificado. Sin embargo, al intentar desplegar el backend y la base de datos, se encontraron limitaciones en las plataformas de despliegue automático, todo debido a que las dependencias que se permiten instalar son muy reducidas o directamente nulas. Después de algunos intentos sin resultados positivos, se decidió cambiar de enfoque, se optó por plataformas más flexibles pero también más complejas de configurar. Aunque llevó más tiempo, se logró alojar tanto la base de datos como la API de la aplicación en *Google Cloud Platform*.

5.2. Ampliación de lo implementado

En caso de querer ampliar la funcionalidad de la aplicación, se contempla la posibilidad de agregar nuevos filtros y formatos para ofrecer a los usuarios una experiencia más personalizada. Para llevar a cabo esta ampliación, sería necesario crear nuevos endpoints en la API de la aplicación. Dependiendo de si se trata de un filtro o un formato, se crearía un nuevo archivo en la ruta *modify_images/filters/* o en la ruta *modify_images/formats/*, respectivamente. En este archivo se especifica la lógica de la implementación de la nueva funcionalidad.

Supongamos que queremos añadir un nuevo filtro. En este caso, se deben importar todas las librerías y funciones necesarias porque lo primero que se realizará será la creación del Blueprint, que nos servirá de organizador y estructurador en la aplicación. A continuación definimos la ruta de acceso a este filtro y finalmente implementamos la funcionalidad en una función. En la figura se puede ver cómo se realiza lo necesario para obtener la imagen del cuerpo de la petición, que debe ser de tipo PATCH tal y como se ha definido.

```
1 import numpy as np
2
3 from flask import Blueprint, jsonify, request
4 from db_connection import get_db_connection
5
6
7 example_bp = Blueprint('example', __name__)
8
9 @example_bp.route('/example', methods=['PATCH'])
10 def example():
11     try:
12         conn = get_db_connection()
13         cursor = conn.cursor()
14
15         data = request.get_json()
16         image = data['image']
17
18         if image == None:
19             return jsonify({"message": "Image not found"}), 404
20
21         # Implementación del código de la nueva funcionalidad
22
23         image = {
24             "image": final_image
25         }
26
27         return jsonify(image), 200
28     except Exception as error:
29         return jsonify({"message": "An error has occurred.", "error": f"{error}"}) , 500
```

Figura 5.1: Conexión a la base de datos desde la API

Posteriormente, se agregaría este nuevo filtro al archivo principal app.py, junto con la URL específica para acceder a esta funcionalidad. Esto se realizaría de la siguiente manera:

```
1 from routes.modify_images.filters.example import example_bp
2
3 app.register_blueprint(example_bp, url_prefix='/api/modify-images')
```

Figura 5.2: Conexión a la base de datos desde la API

Finalmente, se adaptaría la parte visual de esta nueva característica en el frontend de la aplicación.

5.3. Análisis de acierto y error de detección de rostros

Para detectar el índice de acierto que tiene la aplicación sobre la detección de rostros en imágenes, se ha decidido realizar un estudio que analice la implementación de la detección de rostros sobre 120 imágenes aleatorias, dividiendo el conjunto de imágenes en tres grupos: imágenes grandes, medianas y pequeñas. Esta división se debe a que, en la implementación, con el fin de obtener mejores resultados, se decidió dividir la función que detecta rostros por tamaño en píxeles de las imágenes, aplicando distintos coeficientes a la función. Se decidió de tal manera que las imágenes grandes son aquellas que, ya sea en el largo o en el ancho, tengan más de 4.000 píxeles, las imágenes medianas, las que tienen, en el largo o en el ancho, más de 2.000 píxeles y las pequeñas aquellas que se encuentran en el rango de 0 a 2.000 píxeles. Además, para que el estudio sea representativo, se deben evaluar todos los distintos casos que se pueden dar, es por esto que se decidió utilizar el modelo para evaluar el impacto de la clasificación de imágenes llamado *F1 Score in Machine Learning* [16].

A continuación se puede observar un ejemplo de verdadero positivo y un ejemplo de falso positivo, es decir, tras la entrada de una imagen con un rostro, en el primero se detecta correctamente, mientras que en el segundo se detecta incorrectamente, ya sea sin haber encontrado ninguna coincidencia o encontrando coincidencias de más.

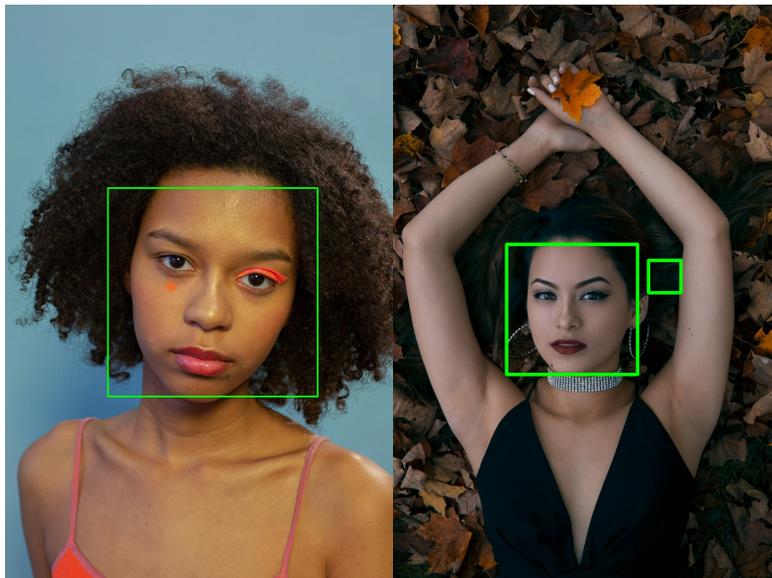


Figura 5.3: Verdadero positivo (izquierda) frente a falso positivo (derecha)

Por otra parte, se pueden observar ejemplos de verdaderos negativos y falsos negativos, es decir, entradas de imágenes sin rostros, que en el primero de los casos no se detecta ningún rostro, como se había planeado, y en el segundo de los casos, se detecta algún rostro.



Figura 5.4: Verdadero negativo (izquierda) frente a falso negativo (derecha)

En la figura 5.5 se representa el estudio realizado sobre las verdaderos positivos. Si el rostro es detectado correctamente se considera un true positive o verdadero positivo (TP), en caso contrario, se considera un false positive o falso positivo (FP).

TRUE POSITIVES								
Pequeñas			Medianas			Grandes		
Calificación	Observaciones		Calificación	Observaciones		Calificación	Observaciones	
0	FP	Un rostro de más	0	FP	Un rostro de más	0	FP	Un rostro de más
1	TP		1	TP		1	FP	Un rostro de más
2	TP		2	FP	Ningún rostro	2	TP	
3	TP		3	TP		3	TP	
4	TP		4	TP		4	TP	
5	TP		5	TP		5	FP	Un rostro de más
6	TP		6	TP		6	TP	
7	FP	Un rostro de más	7	TP		7	TP	
8	TP		8	TP		8	TP	
9	TP		9	TP		9	TP	
10	FP	Ningún rostro	10	TP		10	FP	Un rostro de más
11	TP		11	TP		11	TP	
12	FP	Ningún rostro	12	TP		12	FP	Un rostro de más
13	TP		13	TP		13	TP	
14	TP		14	TP		14	TP	
15	TP		15	FP	Ningún rostro	15	TP	
16	TP		16	TP		16	TP	
17	FP	Ningún rostro	17	FP	Un rostro de más	17	TP	
18	TP		18	TP		18	FP	Un rostro de más
19	TP		19	TP		19	FP	Rostros de más

TOTAL		
FP	5	25%
TP	15	75%

TOTAL		
FP	4	20%
TP	16	80%

TOTAL		
FP	7	35%
TP	13	65%

TOTAL		
FP	16	26.67%
TP	44	73.33%

Figura 5.5: Análisis del índice de acierto de detección de rostros para los verdaderos positivos

A continuación aparece la figura 5.6, que representa el estudio realizado sobre los verdaderos negativos. Si el rostro no es detectado se considera un true negative o verdadero negativo (TN), en caso contrario, se considera un false negative o falso negativo (FN).

TRUE NEGATIVES								
Pequeñas			Medianas			Grandes		
Calificación	Observaciones		Calificación	Observaciones		Calificación	Observaciones	
0	TN		0	TN		0	TN	
1	TN		1	TN		1	TN	
2	TN		2	TN		2	FN	Un rostro
3	TN		3	TN		3	TN	
4	TN		4	TN		4	TN	
5	TN		5	TN		5	TN	
6	TN		6	TN		6	TN	
7	TN		7	TN		7	TN	
8	TN		8	TN		8	TN	
9	TN		9	TN		9	TN	
10	TN		10	TN		10	TN	
11	TN		11	TN		11	FN	Tres rostros
12	TN		12	TN		12	FN	Dos rostros
13	TN		13	TN		13	TN	
14	TN		14	TN		14	FN	Un rostro
15	TN		15	TN		15	FN	Un rostro
16	TN		16	TN		16	TN	
17	TN		17	TN		17	TN	
18	TN		18	TN		18	FN	Un rostro
19	TN		19	TN		19	TN	

TOTAL		
FN	0	0%
TN	20	100%

TOTAL		
FN	0	0%
TN	20	100%

TOTAL		
FN	7	30%
TN	13	70%

TOTAL		
FN	6	10%
TN	54	90%

Figura 5.6: Análisis del índice de acierto de detección de rostros para los verdaderos negativos

Con todos estos resultados, podemos determinar que el modelo destaca en la identificación de verdaderos negativos, logrando detectar 54 de 60 casos correctamente. Este nivel de precisión en la identificación de negativos es excelente y demuestra un gran nivel para detectar casos en los que no aparecen rostros. Para el caso de los verdaderos positivos, aunque no es tan sobresaliente como el de los negativos, sigue siendo muy bueno, consiguiendo detectar los rostros correctamente en 44 casos de 60.

Si profundizamos un poco y dividimos el análisis por el tamaño imagen podemos detectar lo siguiente:

- En el caso de las imágenes pequeñas se detectaron solo 5 casos de error sobre 40 casos totales, la detección de este tipo de imágenes cumple con las expectativas y realiza un buen trabajo, casi perfecto.
- Del mismo modo ocurre con las imágenes de mediano tamaño, donde se detectaron 4 casos de error sobre 40 casos totales. La detección de rostros sobre este tamaño de imagen es aún mejor que el anterior, aspecto que se considera muy relevante ya que las imágenes que más se manipulan en el día a día tienen estas dimensiones.
- Por último aparece el caso de las imágenes grandes, donde en 14 de

los 40 casos se detectó incorrectamente los rostros. Esto sugiere que al tener que analizar un número mucho más elevado de píxeles, se puede llegar a tener mayor índice de error.

A continuación podemos ver la métrica que mide el índice de error y acierto de la detección facial.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} = \frac{44}{44 + 16} = \frac{44}{60} \approx 0.733\ (73.3\%)$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} = \frac{44}{44 + 6} = \frac{44}{50} = 0.88\ (88\%)$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} = 2 \times \frac{0.73 \times 0.88}{0.73 + 0.88} = 2 \times \frac{0.645}{1.613} \approx 0.8\ (80\%)$$

Figura 5.7: Análisis de la métrica utilizada

Con estos resultados podemos destacar tres grandes aspectos:

- Precision (Precisión): El valor obtenido significa que, de todas las imágenes que el modelo etiquetó como positivas, el 73.3 % fueron realmente positivas. Este nivel de precisión indica que el modelo es bastante eficaz para evitar falsos positivos.
- Recall (Sensibilidad): Lo que indica el valor obtenido es que se identificó correctamente el 88 % de todas las imágenes que eran realmente positivas, demostrando que el modelo es muy bueno en la detección de verdaderos positivos.
- F1 Score: Al obtener un 80 %, se demuestra que el modelo mantiene un buen equilibrio entre evitar falsos positivos y no descartar verdaderos positivos.

5.4. Pruebas de usuario

Las pruebas de usuarios en páginas web son una herramienta fundamental para evaluar la usabilidad y la eficacia de un sitio web. Estas pruebas consisten en observar cómo los usuarios reales interactúan con la página, identificando posibles problemas y áreas de mejora. Es por esto que, una vez realizada la página, se procedió a realizar estas pruebas sobre cinco usuarios aleatorios. Las pruebas están compuestas de dos baterías de preguntas, unas para realizarlas antes de navegar por la página y otra durante la propia navegación.

Las preguntas previas son las siguientes:

1. ¿Cuál es tu edad? ¿Estudias o trabajas?
2. ¿Con qué frecuencia utilizas internet? ¿Para qué fines lo utilizas?
3. ¿Has utilizado alguna vez una aplicación o página web de edición de imágenes? ¿Qué te gustó o no te gustó de esa experiencia?
4. ¿Qué tipo de dispositivos utilizas para acceder a internet?
5. ¿Con qué frecuencia editas imágenes?
6. ¿Te gusta experimentar con nuevas aplicaciones de edición de imágenes?
7. ¿Qué tipo de funcionalidades destacarías más en una página de edición de imágenes?
8. ¿Qué te motiva a editar imágenes normalmente?
9. ¿Cuánto tiempo estás dispuesto a invertir en la edición de imágenes?

Por otro lado aparecen las preguntas que se deben responder durante la navegación de la página, tienen un fin más técnico y se encuentran divididas en criterios, concretamente en cinco criterios:

Sobre la **facilidad de uso**:

- ¿Cómo calificarías la facilidad de uso del sitio para editar imágenes?
- ¿Te resultó fácil encontrar las funcionalidades que estabas buscando en el sitio?
- ¿Cómo describirías tu experiencia general utilizando la página?

Sobre el **aspecto**:

- ¿Te gustó el diseño general del sitio y las características que ofrece?
- ¿Te gustó la disposición de los elementos en la página?
- ¿Encontraste que el sitio estaba organizado de manera lógica y fácil de entender?

Sobre el **información de la página**:

- ¿Consideras sencillo el proceso de reportar un error?
- ¿Crees que la página deja claro en todo momento qué hace cada filtro y formato?
- ¿Crees que la página deja claro el objetivo principal?

Sobre la **contenido de la página**:

- ¿Encuentras que el sitio satisface tus necesidades de edición de imágenes?
- ¿Crees que existe algún grupo de usuarios a los que no se les proporciona suficiente contenido específico?
- ¿Consideras que la página puede ayudarte en tu día a día?

Sobre la **ejecución de la página**:

- ¿Hubo algún problema técnico o de navegación que encontraste durante el uso del sitio?
- ¿Crees que el sitio es accesible para usuarios con diferentes niveles de habilidades informáticas?
- ¿Crees que los tiempos de carga son adecuados?

Para poder evaluar correctamente todas las respuestas se ha decidido recoger la información del usuario en la siguiente tabla, donde debe establecer una puntuación del uno al cinco para cada pregunta de cada criterio, donde el uno simboliza *en desacuerdo* y cinco simboliza *de acuerdo*.

Criterio	1	2	3	4	5
Uso					
Aspecto					
Información					
Contenido					
Ejecución					

Tabla 5.1: Pruebas de usuario

5.4.1. Información referente al usuario 1

1. 21. Estudio.
2. Aproximadamente entre 2-3 horas al día, fines educativos y de entretenimiento.
3. Si, páginas para convertir el formato de imágenes y apps de edición fotográfica.
4. Móvil y ordenador.
5. Aproximadamente una vez a la semana.
6. Si, me resulta interesante ver qué se puede lograr.
7. Que los filtros que se puedan aplicar sean variados y que conste de herramientas para organizar capas en la misma foto.
8. Realizar retoques en las mismas.
9. El necesario, puede ir desde cinco minutos a una hora.

Criterio	1	2	3	4	5
Uso					X
					X
					X
Aspecto					X
					X
					X
Información					X
					X
					X
Contenido					X
					X
					X
Ejecución					X
					X
	X				

Tabla 5.2: Respuestas del Usuario 1

Tabla 5.3: Respuestas del Usuario 1

5.4.2. Información referente al usuario 2

1. 22. Estudio.
2. Para búsquedas de información y para cambios de formatos de imágenes.
3. Sí. La facilidad de uso.
4. Móvil y iPad.
5. Aproximadamente una o dos veces a la semana.
6. Sí.
7. Buenos filtros, buena resolución.
8. Sacar lo máximo a la imagen.
9. Bastante tiempo, no me importa.

Criterio	1	2	3	4	5
Uso				x	
					x
				x	
Aspecto		x			
			x		
				x	
Información					x
					x
					x
Contenido					x
	x				
				x	
Ejecución				x	
				x	
					x

Tabla 5.4: Respuestas del Usuario 2

5.4.3. Información referente al usuario 3

1. 22. Estudio.
2. Mucho. Para fines sociales o entretenimiento y para estudiar.
3. Rara vez, no me gusta editar fotos.
4. Móvil y ordenador.
5. Poca.
6. No mucho.
7. Que sea sencilla e intuitiva para que todo el mundo la use.
8. No edito imágenes normalmente.
9. Sinceramente poco.

Tabla 5.5: Respuestas del Usuario 3

Criterio	1	2	3	4	5
Uso					X
					X
					X
Aspecto					X
					X
					X
Información				X	
			X		
					X
Contenido					X
					X
			X		
Ejecución			X		
					X
				X	

5.4.4. Información referente al usuario 4

1. 48. Trabajo.
2. Todos los días. Para consultar dudas del día a día.
3. No.
4. Móvil.
5. Nunca.
6. No.
7. Que sea sencilla de usar.
8. Nada.
9. Lo menos posible.

Criterio	1	2	3	4	5
Uso					X
					X
					X
Aspecto					X
					X
					X
Información					X
					X
					X
Contenido					X
					X
					X
Ejecución					X
				X	
			X		

Tabla 5.6: Respuestas del Usuario 4

5.4.5. Información referente al usuario 5

1. 22. Estudio y trabajo.
2. Todos los días, principalmente para resolver dudas o problemas. Buscar información, imágenes, hacer diapositivas, trabajar, etc.
3. Si, para eliminar el fondo o modificar el color y recortar imágenes. En ocasiones aparecen problemas para descargar el resultado.
4. Móvil, portátil o tablet.
5. Cada vez que necesito hacer unas diapositivas o informes, digamos que dos o tres veces al mes, pero en ocasiones mucho más.
6. Cuando tengo buenos resultados con una aplicación, intento usarla siempre, pero en ocasiones pruebo nuevas opciones.
7. Que describan bien lo que se puede hacer.
8. Principalmente por trabajo o estudio.
9. No más de 30 minutos por imagen, depende del objetivo final.

Criterio	1	2	3	4	5
Uso				X	
				X	
			X		
Aspecto					X
					X
					X
Información					X
					X
					X
Contenido		X			
					X
					X
Ejecución					X
				X	
				X	

Tabla 5.7: Respuestas del Usuario 5

5.4.6. Interpretación de las pruebas de usuario

En conclusión podemos sacar varios puntos a través de las pruebas realizadas:

- Los tiempos de espera es el aspecto que más en cuenta tienen los usuarios a la hora de valorar una página. Los resultados referentes a estos tiempos no son muy favorables y es por esto que se decidió mejorar este aspecto para lograr que los usuarios que usen la página no tengan que esperar tanto por sus resultados, para ello se aumentaron los recursos de la máquina virtual en la que se encontraba alojado el código, suponiendo un aumento en los costos asociados.
- La información referente a cada formato y cada filtro se detectó que era insuficiente, es por esto que se decidió mejorar las definiciones, siendo más explícito.
- El acceso a la página desde donde se puede conseguir el contacto con el desarrollador era poco accesible y resultaba difícil de encontrar, por lo que se colocó en la barra lateral para que esté al alcance desde cualquier página.
- El contenido para todo tipo de público parece ser insuficiente, uno de los apartados que se proponen como un aspecto a mejorar es el hecho de hacer completamente accesible la página, para todo tipo de público.

En el gráfico de la figura 5.8, se tienen las medias de los cinco usuarios para cada pregunta de cada criterio, además de la media para cada criterio en general. Tal y como se puede apreciar, la media general por pregunta es favorable, a excepción del criterio de ejecución, que tiene una media general considerablemente menor a las demás. En general, las pruebas de usuario dieron un resultado satisfactorio, ninguno de los criterios tiene una media inferior a cuatro puntos.

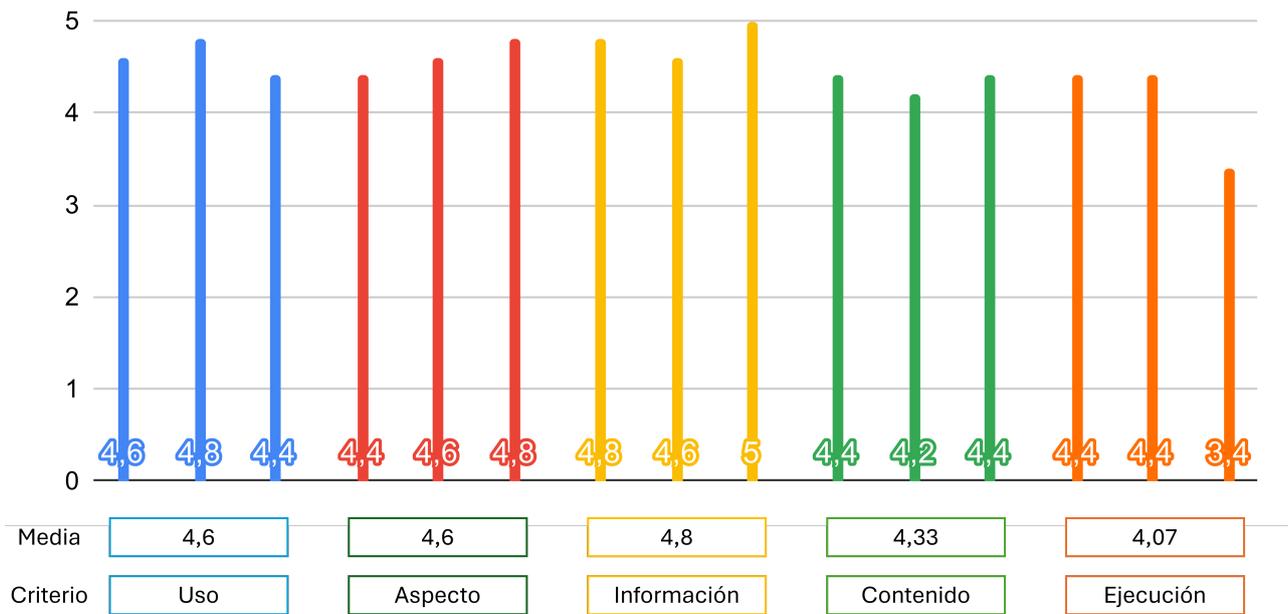


Figura 5.8: Gráfica que representa la media por pregunta por criterio

Capítulo 6

Conclusiones y problemas encontrados

Se ha desarrollado una aplicación que permite automatizar la personalización y estandarización de imágenes para la publicación de las mismas en distintas plataformas. Con esto se logra que el usuario ahorre tiempo y esfuerzo para adaptar las imágenes a los distintos contextos que se tratan en la aplicación. Se intentó abarcar el mayor número de público posible mediante el diseño intuitivo y sencillo de la aplicación, logrando que personas sin conocimientos de edición de imágenes puedan adaptar sus imágenes a los distintos escenarios contemplados en la realización del proyecto.

6.1. Límites

En cuanto a los límites del proyecto, existen diversas áreas en las que se podría mejorar y expandir las funcionalidades:

- Ante recursos ilimitados, se podría implementar una base de datos sin restricciones en cuanto al número de imágenes que cada usuario puede almacenar.
- La plataforma siempre estará abierta a la inclusión de filtros y formatos nuevos. Ante cualquier inserción, se debe evaluar el impacto de cada nueva funcionalidad.
- Se podría considerar la implementación de un foro dentro de la plataforma donde los usuarios puedan aportar ideas de nuevos formatos y filtros. Estas sugerencias podrían estar sujetas a un sistema de puntuación, donde las más populares sean consideradas para su desarrollo futuro.

6.2. Problemas encontrados

Para definir todos los problemas encontrados hay que enfocarse en las dificultades sobre la implementación.

- El primer obstáculo fue la pérdida de acceso a la máquina virtual proporcionada por la universidad. Aunque el código estaba en un repositorio privado de GitHub y no se perdió, se invirtió mucho tiempo en encontrar una alternativa.
- La necesidad de suficiente almacenamiento para alojar un gran número de imágenes en la base de datos fue otro problema. Ninguna plataforma ofrecía paquetes ideales. Tras investigar, se optó por Google Cloud Platform, que ofrecía créditos gratuitos para nuevos usuarios, permitiendo desplegar una base de datos adecuada para la aplicación.
- El despliegue del backend fue complicado debido a la imposibilidad de instalar dependencias por falta de espacio en las alternativas disponibles. Finalmente, gracias a Google Cloud Platform, se pudo instalar toda la aplicación sin mayores inconvenientes.

6.3. Relación con materias cursadas

En este apartado se destacarán las similitudes entre las asignaturas vistas en la carrera y el desarrollo de este proyecto final.

- *Desarrollos de Sistemas Informáticos y Sistemas y Tecnologías Web*, conceptos básicos como la definición de una API y sus endpoints, uso de peticiones HTTP y sus verbos, autenticación de usuarios mediante tokens, y la conexión entre diferentes capas de la aplicación.
- *Gestión de Proyectos Informáticos*, metodologías de gestión de proyectos como historias de usuario y sprints semanales, permitiendo una planificación efectiva y un enfoque iterativo en el proyecto.
- *Base de Datos y Administración y Diseño de Base de Datos*, comprensión y aplicación de operaciones básicas de bases de datos, realización de consultas, diseño de modelos de datos y sus relaciones.
- *Sistemas de Interacción Persona-Computador y Usabilidad y Accesibilidad*, diseño centrado en el usuario y control de la gama de colores para la captación de la atención de los usuarios.
- *Inteligencia Emocional*, control de emociones y pensamientos, crucial para manejar el estrés y los desafíos durante la realización del proyecto.

Capítulo 7

Summary and Conclusions

An application has been developed that allows for the automation of image personalization and standardization for publication on various platforms. This helps users save time and effort when adapting images to different contexts addressed by the application. An effort was made to reach the widest possible audience through the intuitive and simple design of the application, allowing people without image editing knowledge to adapt their images to the various scenarios considered in the project's implementation.

7.1. Limits

Regarding the project limits, there are various areas where functionalities could be improved and expanded:

- With unlimited resources, a database could be implemented without restrictions on the number of images each user can store.
- The platform will always be open to the inclusion of new filters and formats. For any new additions, the impact of each new functionality should be evaluated.
- The implementation of a forum within the platform could be considered, where users can contribute ideas for new formats and filters. These suggestions could be subject to a rating system, where the most popular ones are considered for future development.

7.2. Problems found

To define all the problems found, one must focus on the difficulties regarding implementation.

- The first obstacle was the loss of access to the virtual machine provided

by the university. Although the code was in a private GitHub repository and was not lost, a lot of time was spent finding an alternative.

- The need for sufficient storage to host a large number of images in the database was another problem. No platform offered ideal packages. After investigating, Google Cloud Platform was chosen, which offered free credits for new users, allowing the deployment of an adequate database for the application.
- The deployment of the backend was complicated due to the inability to install dependencies due to space limitations on the available alternatives. Finally, thanks to Google Cloud Platform, the entire application could be installed without major inconveniences.

7.3. Relationship with subjects studied

In this section, the similarities between the subjects studied in the degree and the development of this final project will be highlighted.

- *Development of Information Systems and Web Systems and Technologies*, basic concepts such as the definition of an API and its endpoints, use of HTTP requests and their verbs, user authentication using tokens, and the connection between different layers of the application.
- *Management of Information Systems Projects*, project management methodologies such as user stories and weekly sprints, allowing effective planning and an iterative approach to the project.
- *Databases and Database Administration and Design*, understanding and applying basic database operations, performing queries, designing data models and their relationships.
- *Human-Computer Interaction Systems and Usability and Accessibility*, user-centered design and control of the color scheme to capture the attention of users.
- *Emotional Intelligence*, control of emotions and thoughts, crucial for managing stress and challenges during the project execution.

Capítulo 8

Presupuesto

La tabla muestra el desglose de horas y costos para las diferentes actividades realizadas. En este caso, los precios están ajustados para reflejar el contexto de España, donde los costos tienden a ser más bajos en comparación con otros países.

- **Análisis y estudio**, evaluación de requisitos y objetivos del proyecto.
- **Base de datos**, diseño y modelado de la estructura de datos.
- **Backend**, desarrollo de la lógica y funcionalidades del servidor.
- **Frontend**, creación de la interfaz de usuario.
- **Memoria**, redacción de la memoria final del proyecto.

El total del proyecto asciende a 2.225 euros, considerando las horas invertidas y los precios por hora establecidos para cada actividad.

Actividad	Horas	Precio por hora	Total
Análisis y estudio	10	10€	100€
Base de datos	5	20€	100€
Backend	35	25€	875€
Frontend	50	20€	1.000€
Memoria	20	7.5€	150€
Total	120		2.225€

Tabla 8.1: Resumen del presupuesto

Bibliografía

- [1] Del Valle, J. M. R. V., & Granados, J. M. (2007). Programación en capas. *Di Mare, Costa Rica*.
- [2] Trigás Gallego, M. (2012). Metodología Scrum
- [3] Guevara, M. L., Echeverry, J. D., & Urueña, W. A. (2008). Detección de rostros en imágenes digitales usando clasificadores en cascada. *Scientia et technica*, 1(38).
- [4] Catalán Vitas, D. (2017). Detección automática de personas mediante Histograma de Gradientes Orientados.
- [5] Artola Moreno, Á. (2019). Clasificación de imágenes usando redes neuronales convolucionales en Python.
- [6] Santiago Rojo, J. E. D. (2018). *Detección de vehículos en entornos multi-cámara utilizando información contextual (Bachelor's thesis)*.
- [7] Sharma, A., Pathak, J., Prakash, M., & Singh, J. N. (2021, December). Object detection using OpenCV and python. In *2021 3rd international conference on advances in computing, communication control and networking (ICAC3N) (pp. 501-505)*. IEEE.
- [8] New Layer. (20 de Noviembre de 2019). RESHAPE ANY FACE in Photoshop (Face-Aware Liquify Tool)
- [9] Ginestà, M. G., & Mora, O. P. (2012). Bases de datos en PostgreSQL
- [10] González Duque, R. (2011). Python para todos
- [11] Dwyer, G. (2016). *Flask By Example*. Packt Publishing Ltd.
- [12] Boduch, A. (2017). *React and react native*. Packt Publishing Ltd.
- [13] Pérez, J. E. (2019). Introduccion a JavaScript.
- [14] Muyón, C., & Montaluisa, F. (2020). Métodos de seguridad de la información para proteger la comunicación y los datos de servicios web REST en peticiones HTTP utilizando JSON Web Token y Keycloak Red Hat Single Sign On. *Revista Ibérica de Sistemas e Tecnologias de Informação*, (E29), 198-213.
- [15] Cloud, S. Q. L. (2018). Google Cloud Platform.
- [16] Rohit Kundu. (16 de Diciembre de 2022). F1 Score in Machine Learning: Intro & Calculation